

**Korea patent office (KR) Unexamined Patent
Publication(A)**

KOREAN

(51) Int.Cl. H04R 1/00

Publication No 10-2000-0073035

Publication Date 2000-12-05

Application No 10-1999-0016055

Application Date 1999-05-04

Agent Seung-Yeong Moon Inventor

Yong-Gil Yoon

Sang-Yeong Jeong

Applicant SHIN YONG KYU CO., LTD

java.lang.NullPointerException

(54) device for to regulate an angle of speaker**Abstract**

The present invention relates to the angle adjusting of a speaker, and this the present invention draws data corresponding to the music mode selected in the music mode selecting unit than from the stored data and rotates the speaker rotating unit at the angle of data drawn out and it controls the angle of a speaker to the music mode.

In this way, by doing a speaker can be controlled by the angle in which a user wants. The stereosound (surround) of the tone quality is increased and the music of the lively tone quality can be listened to

**Representative Drawing(s)****Description****Brief Explanation of the Drawing(s)**

Fig. 1 is a block diagram showing a configuration of the hitherto audio system

Fig. 2 is a cross-sectional view showing a structure of the conventional speaker

Fig. 3 is a block diagram of the tilting device of a speaker

Fig. 4 is a cross-sectional view showing a structure of a speaker

Fig. 5 is a flowchart showing the angle control state of the speaker in the present invention.

■ Details of the Invention

■ Purpose of the Invention

- The Technical Field to which the Invention belongs and the Prior Art in that Field

The present invention relates to an audio, particularly, to the tilting device of the speaker which automatically controls the speaker angle for the presentable of the lively tone quality according to the music mode user chooses and listens to the lively music.

Generally, in the vehicle or the performance place etc, the audio system is the instrument outputting or the sound. This is comprised of the music mode selecting unit (11) as shown in Fig. 1, selecting mode, the storage (12), the MICOM (13) drawing out signal processing data corresponding to the mode selected in the music mode selecting unit (11) than from the storage (12), the signal processing unit processing the dependence input audio signal in signal processing data according to the music mode in the MICOM (13), the amplifier (15) amplifying the audio signal processed at the signal processing unit and the speaker (16) outputting the amplified audio signal to the amplifier (15) to an outside. As to the storage (12), signal processing data corresponding to each music mode is stored.

In this way, in a user is the music mode selecting unit (11), if the comprised hitherto audio system is ballad, the MICOM (13) draws out data for processing the ballad signal among signal processing data from the storage (12). The comprised hitherto audio system delivers this to the signal processing unit (14).

According to the ballad mode process data, the signal processing unit (14) processes the input audio signal. After the processed audio is amplified in the amplifier (15), it is outputted through the speaker (16) to the outside.

Here, as to the installation structure of the conventional speaker, as shown in Fig. 2, the speaker (16) is fixed to the vehicle body pannel (10) of a vehicle to the screw (18). It is installed so that the speaker grill (17) of the speaker (16) from any kind of impact be separated with fixed interval.

There is a problem that as to such speaker but, the capability which delivers the tone quality various since being fixed to the vehicle vehicle body pannel is lowered. And the stereoscopic effect (surround effect) of the tone quality is degraded.

- The Technical Challenges of the Invention

Thus, the present invention is comprised of consideration of the conventional fundamental fault as shown in the above. It provides the speaker tilting device which controls the angle of a speaker according to the music mode and which implements the tone quality various and lively and can listen, it has the purpose.

The present invention for achieving this purpose is done by a feature to comprise the slide part which slides according to the curved surface in case a speaker is rotated with each music mode and the music mode selecting unit: selecting the DSP mode and the storage storing angle data corresponding to each music mode and MICOM which draws out angle data corresponding to the music mode selected in the music mode selecting unit than from the storage and which MICOM output and the motor driver which outputted in MICOM power in order to rotate a speaker at the designated angle corresponding to angle data and the speaker unit: receiving the power delivered from the motor driver and rotates a speaker and speaker rotating unit.

■ Structure & Operation of the Invention

Hereinafter, the embodiment is circumstantially illustrated based on the attached drawing. Fig. 3 is a plan diagram of the tilting device. Fig. 4 is a cross-sectional view showing a structure of a speaker.

The present invention is comprised of as shown in Fig. 3, each music mode and the music mode selector (21) selecting the DSP mode, the storage (22) storing angle data corresponding to each music mode, the MICOM (23) drawing out angle data corresponding to the music mode selected in the music mode selector (21) than from the storage (22) and outputs, the motor driver (24), the speaker rotating unit (25) receiving power delivered from the motor driver (24) and rotates the speaker (29), the slide part (27), the amplifier amplifying the audio signal processed at the signal processing unit (27), and the speaker (29) outputs the amplified audio signal to the amplifier (28) to an outside. The motor driver (24) outputted in the MICOM (23) delivers the power in order to rotate the speaker (29) at the designated angle corresponding to angle data drawn out from the storage (22). The slide part (27) slides the speaker (29) according to the curved surface of the speaker rotating unit (25) and the signal processing unit (27) processing the dependence input audio signal processing data according to the music mode outputted in the MICOM (23).

As to the speaker rotating unit (25), the motor (251) which operates if the power source is applied and the driver (24) is comprised. The driving gear (252) and the slave gear (253) delivering the power are connected by the operation of the motor (251) to the tooth form and the operation is comprised. And the belt (254) connecting the motor (251) and driving gear (252) is comprised.

As to the slide part (26), the first curved surface (261a) which forms the tooth form in order to go in and out between the slave gear (253) of the speaker rotating unit (25) is equipped. The second curved surface (261b) which is bent so that the other side be slid is equipped. And the drive material (261) in which the fragment (261c) in which the screw hole is formed is equipped in an inside outside the first, and the second curved surface (261a) (261b) is comprised. The third curved surface (262a) which is bent in order to go in and out between the first, and the second curved surface (261a) (261b) while surrounding the drive material (261) is equipped. The fixing member (262) in which the fixture piece (262b) in which the screw hole is formed is equipped inside is comprised. And the screw (263) fixing the gall rim fragment (261c) and speaker and the fixing member (262) and vehicle body pannel (10) is comprised. One side is installed so that the speaker grill (29) of the speaker (29) from the minute impact be located in one side of the drive material (261) with fixed.

If the action which becomes with this and *** is illustrated, as shown in Fig. 5, if an audio is lit, it confirms whether it is the DSP mode (S31). The output (S40) is to the speaker (29) if it is not confirmation result DSP mode. If it is the confirmation result DSP mode, it again confirms whether the sound field mode is the hall mode (S32). Speaker angle data of the odd mode state is drawn out from the storage (22) (S33) if the confirmation result sound field mode is the odd mode. If it is not odd mode, the confirmation result sound field mode again confirms whether the sound field mode is the exhortation (church) mode (S34). Speaker angle data of the exhortation mode stored in the storage (22) is drawn out (S35) if the confirmation result sound field mode is the exhortation mode. If it is not exhortation mode, the confirmation result sound field mode again confirms whether a location is the driver mode (S36). Speaker angle data of the driver mode stored in the storage (22) is drawn out (S37) if the confirmation result location is the driver mode. The confirmation result location is the return (return) if it is not driver mode.

The MICOM (23) as described in the above runs the motor driver (24) with speaker angle data drawn out from the storage (22). The motor driver (24) is generated the power (S38) rotating the speaker (29). The speaker rotating unit (25) rotates the speaker (29) at the power generated in the motor driver (24) according to each speaker angle data.

And then, the speaker (29) outputs an audio to the target direction (S40). And a user therefore listens to music having with presence.

■ Effects of the Invention

As described in the above, the present invention has the effect that angle data corresponding to the music mode are drawn out from the storage. The angle of a speaker can be controlled according to the music mode by running the speaker rotating unit to data drawn out.

Moreover, as described in the above, according to the music mode, the angle adjusting of a speaker is possible. Therefore, a speaker has the advantage which it plans in order to increase the stereoscopic (surround) of the tone quality and it listens to the music of the lively tone quality.



Scope of Claims

Claim[1] :

The tilting device of a speaker of the audio reproducing system, wherein in case a speaker is rotated in the music mode and the music mode selecting unit (21): selecting the DSP mode and the storage (22) stores angle data corresponding to each music mode and the MICOM (23): which draws out angle data corresponding to the music mode selected in the music mode selecting unit (21) than from the storage (22) and outputs the angle data to the motor driver (24): which outputted in the MICOM (23) delivers the power in order to rotate the speaker at the designated angle corresponding to angle data and the speaker rotating unit (25): receiving the power delivered from the motor driver (24) and rotates the speaker (29) and speaker rotating unit (25), a slide part (26) which slide is comprised according to the curved surface.

Claim[2] :

The tilting device of a speaker of claim 1, wherein in the speaker rotating unit (25), the motor (251) operates if the power source is applied as the motor driver (24) is comprised; the driving gear (252) and slave gear (253) delivering the power are combined with the operation of the motor (251) to the tilting operation is comprised; and the belt (254) connecting the motor (251) and driving gear (252) is comprised.

Claim[3] :

The tilting device of a speaker of claim 1, wherein in the slide part (26), the first curved surface (261a) forms the tooth form in order to go in gear between the slave gear (253) of the speaker rotating unit (25) equipped; the second curved surface (261b) which is bent so that the other side be slid is equipped in the slide material (261) in which the gear rim fragment (261c) in which the screw hole is formed is equipped in the slide material (261) outside the first, and the second curved surface (261a) (261b) is comprised; the third curved surface (261c) which is bent in order to be slid the first, and the second curved surface (261a) (261b) while surround the drive material (261) is equipped; the fixing member (262) in which the fixture piece (262b) in which the screw hole is formed is equipped in an inside is comprised; and it comprises the speaker grill (291) protecting the speaker (29) from the minute impact in one side of the drive material (261).

THIS PAGE BLANK (USPTO)

(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(51) Int. Cl.⁶
G06F 15/16

(11) 공개번호 특1999-016055
(43) 공개일자 1999년08월05일

(21) 출원번호	특1997-038477
(22) 출원일자	1997년08월12일
(71) 출원인	한국전자통신연구원 정선중 대전광역시 유성구 가정동 161번지
(72) 발명자	한우중 대전광역시 유성구 어은동 한빛아파트 102-703 박경 대전광역시 서구 월평동 주공아파트 210-1406 윤석한 대전광역시 유성구 어은동 한빛아파트 108-1601
(74) 대리인	신영무, 최승민

실사청구 : 있음

(54) 다수의 주프로세서 및 보조 프로세서를 갖는 프로세서의 구조 및 보조 프로세서 공유 방법

요약

본 발명은 다수의 주프로세서 및 보조 프로세서를 갖는 프로세서의 구조 및 보조 프로세서 공유 방법에 관한 것으로, 보조 프로세서 명령 수행시 예외 상황이 발생한 경우에는 대피 레지스터 화일을 이용하여 주프로세서가 보조 프로세서 명령을 입력할 때 레지스터 화일 상태를 보존하고 이후에 보조 프로세서로부터 예외 상황 발생 통보가 오면 보존된 상태로 되돌리는 방법을 사용하였다. 또한 상대적으로 작고 주프로세서가 빈번하게 사용되는 일차 캐시 사용에 따라 캐시 효율이 저하되는 문제점을 해결하기 위하여 일차 캐시 바이패스 기능을 제안하고, 주프로세서가 보조 프로세서로 명령어를 전송할 경우 수행 속도가 저하되는 문제점을 해결하기 위하여 별도의 레지스터 화일을 제공하여 프로세서의 병렬성을 높이고 효율성을 향상시킬 수 있는 다수의 주프로세서 및 보조 프로세서를 갖는 프로세서의 구조 및 보조 프로세서 공유 방법이 제시된다.

도면도

도2

명세서

도면의 간단한 설명

- 도 1은 본 발명이 적용되는 보조 프로세서 공유 개념을 설명하기 위한 블록도.
- 도 2는 본 발명에 따른 프로세서 구조의 상세도.
- 도 3은 다수의 명령어 큐를 대상으로 하는 명령어 스케줄링 개념을 설명하기 위한 블록도.
- 도 4는 보조 프로세서 공유를 위한 데이터 인출/저장 방법을 설명하기 위한 블록도.
- 도 5는 대피 레지스터의 사용 방법을 나타낸 흐름도.
- 도 6은 보조 프로세서 예외 상황 처리 과정을 나타낸 흐름도.

*도면의 주요 부분에 대한 부호의 설명

- 11 : 프로세서
- 12 : 내부 연결망
- 13 : 외부 메모리
- 101-1 내지 101-n : 주프로세서
- 102-1 내지 102-m : 보조 프로세서
- 201 : 명령어 캐시/메모리 관리기
- 202 : 명령어 인출 및 검사 유닛
- 203 : 명령어 디코딩 유닛
- 204 : 주프로세서의 명령어 미수 유닛
- 205 : 레지스터 화일
- 206-1 내지 206-m : 실행 유닛
- 207 : 저장/인출 유닛
- 208 : 명령어 큐

- 209 : 보조 프로세서의 명령어 미수 유닛
 210 : 보조 프로세서 레지스터 화일 211-1 내지 211-k : 실행 유닛
 31-1 내지 31-n : 명령어 큐 32 : 명령어 스케줄러
 41-1 내지 41-n : 보조 프로세서 레지스터 파일
 42-1 내지 42-n : 주프로세서
 43-1 내지 43-n : 저장/인출 유닛
 44-1 내지 44-n : 캐시 메모리 45 : 외부 인터페이스

발명의 상세한 설명

발명의 목적

발명이 속하는 기술분야 및 그 분야의 종래기술

본 발명은 다수의 주프로세서 및 보조 프로세서를 갖는 프로세서의 구조 및 보조 프로세서 공유 방법에 관한 것이다.

종래의 프로세서 구조에서 보조 프로세서를 지원하는 방식은 크게 외부 프로세서로 구현하는 방식과 내부 실행 유닛으로 구현하는 방식이 있다. 외부 프로세서로 구현하는 방식은 프로세서 설계는 간단하지만 보조 프로세서 명령어 외부 칩에서 이루어져야 하므로 수행 성능이 떨어지고 주프로세서도 보조 프로세서와 병렬 수행이 되지 않아 주프로세서의 효율도 수행 성능이 떨어진다는 단점이 있다. 또한 내부 실행 유닛으로 구현하는 방식은 주프로세서 내부에서 주프로세서 수행 유닛의 일부처럼 동작할 수 있으므로 수행 성능은 뛰어나지만 복수의 주프로세서가 있는 경우 보조 프로세서 기능이 공유될 수 없다.

반도체 기술의 발전에 따라 프로세서 소자가 수용할 수 있는 회로 또는 트랜지스터의 수가 점점 방대해지고 있으며 이에 따라 복수의 주프로세서를 한 프로세서에 집적시킬 수 있게 되었고, 여러 가지 복잡한 기능들을 프로세서 내부에 집적하게 되었다. 현재 널리 쓰이는 프로세서들은 내부에 작은 캐시 메모리와 하나 또는 두개의 보조 프로세서를 내장하고 있으며 주프로세서의 수행 유닛들도 여러 개를 내장하고 있다. 이와 같이 수행 유닛이 여러 개 있는 경우 이들을 최대한 동시 수행시킴으로써 프로세서 성능을 극대화 할 수 있으며, 이것은 명령어 수준 병렬성(instruction-level parallelism) 정도로서 나타낼 수 있다. 명령어 수준 병렬성을 높이기 위하여 되도록 복수 수행 유닛들간 수행 의존성(dependency)을 없애거나 의존성을 효과적으로 해결하는 방법들이 제시되었다. 이 때 일반적으로 부동 소수점 연산기나 그래픽 처리기 등 보조 프로세서는 수행 시간과 데이터 형식, 수행 명령어 형태 등에 있어서 주프로세서와는 다른 특성을 나타내므로 주프로세서의 복수 수행 유닛들과는 달리 취급되며 이에 따라 보조 프로세서를 포함한 명령어 수행 병렬성이 보조 프로세서 명령어를 포함하는 많은 응용 프로그램의 수행 성능에 중요한 변수가 된다.

복수의 주프로세서를 하나의 프로세서에 집적시킬 경우 사용 빈도가 주프로세서만큼 크지 않은 보조 프로세서를 모든 주프로세서가 따로 갖는 것은 심각한 낭비가 될 수 있으며 특히 보조 프로세서 기능이 일반적인 부동 소수점 연산 기능 외에 그래픽 처리 기능, 통신 처리 기능 등으로 다양해지면 문제가 점점 심각해진다. 즉, 현재의 기술은 복수의 주프로세서를 갖는 상황에서 다양한 보조 프로세서 기능을 효과적으로 지원하기 어려운 문제점이 있다.

종래의 이러한 문제점을 해결하기 위하여 여러 가지 연구가 진행되어 왔으며 그 내용은 다음과 같다.

먼저, 주프로세서에 하나 이상의 보조 프로세서를 외부 버스 인터페이스를 이용하여 연결하는 방법이다. 이 방법은 보조 프로세서만을 위한 특별한 신호 부단을 줄일 수 있는 효과가 있지만 보조 프로세서 명령어를 수행 중인 동안 주프로세서가 다른 일을 수행할 수 없고, 보조 프로세서에 필요한 데이터를 명령어와 같은 방식으로 전달하므로 수행 성능이 저하되어 명령어 수준의 병렬성을 살릴 수 없는 문제점이 있다.

종래에는 또한 보조 프로세서 명령 수행시 발생하는 예외 상황을 최소한의 정보로 처리 및 복구하기 위하여 다음과 같은 방법을 사용하였다. 즉, 예외 처리 후 복구시 돌아갈 위치 정보를 프로그램 카운터가 저장된 큐의 위치만으로 한정하여 전달함으로써 유지할 정보의 양을 줄이는 방법인데, 이 경우 복구시 레지스터의 값 등이 복구되지 않을 수도 있는 문제점이 있다. 또한 저장할 정보는 파이프라인 단계만큼만 저장하고 복구할 수 있고 현재 명령어 수행이 끝나기 전까지 발생하는 예외 상황에만 대처할 수 있으며 다음 명령어가 수행되기 시작하면 복구가 불가능해진다. 이에 따라 보조 프로세서는 주프로세서의 파이프라인과 동기되어 있거나 보조 프로세서가 동작 중인 경우 주프로세서는 정지되어야 하는 문제점이 있다.

명령어 인출 시간을 단축시키기 위하여 부분 디코딩을 사용하는 방법이 있다. 이 방법은 분기를 갖는 명령어 흐름에서 명령어 인출 지연 시간을 줄이는 방법으로, 분기에 대비한 명령어를 미리 인출하므로 보조 프로세서의 성능을 향상시킬 수 있는 효과가 있다. 그러나 보조 프로세서용 명령어가 미리 인출될 가능성이 있다는 점 이외에 보통 흐름에서는 아무 효과가 없으며 보조 프로세서를 위한 직접 연결 방식을 지원하지 않는다.

외부 입출력이나 메모리 접근 동작을 독립적으로 수행하도록 하기 위하여 외부 인터페이스에 디커플 방식 적용한 방법이 있다. 이 방법은 다중 프로세서 환경에서 외부 디바이스에 효율적으로 접근하는 방법이지만 다수의 프로세서가 하나의 보조 프로세서를 명령어 단위로 공유할 경우 효율성을 높이는 방법을 제시해 주지는 않는다. 또한 종래의 내부 수행 유닛들간의 디커플 개념은 다중 프로세서 구조를 포함하지 않으며 보조 프로세서 인터페이스 개념이 없기 때문에 보조 프로세서 예외 상황에 대한 대책이나 보조 프로세서 명령어 스케줄링 방법에 대한 내용이 없다. 그리고 이 방법에 적용되는 큐는 일반적인 선입선출

(FIFO)형태로서 만족되지만 다수의 엔트리를 한번에 검토하고 스케줄링할 수 있는 구조를 제공하지 못하였다.

중래의 파이프라인을 갖는 프로세서에서 부동 소수점 연산기라는 보조 프로세서를 효과적으로 활용한 방법은 주프로세서와 보조 프로세서간 명령어 수준 병렬성을 향상시키는 기법을 기술하고 있으나 큐를 이용한 디커플링 개념을 포함하고 있지 않으며 다수의 주프로세서가 보조 프로세서 큐를 이용한 공유 개념을 제공하지 못하였다. 또한 내부 레지스터에 데이터를 저장함에 있어서 수행될 또는 수행 중인 명령어간의 의존성을 조사하여 데이터 일관성을 유지하도록 하고 있으나 보조 프로세서에 주프로세서당 하나씩, 복수의 레지스터 화일을 두는 방식은 포함되어 있지 않다. 그 외에도 보조 프로세서용 데이터가 작은 내부 캐시를 바이패스하는 방식도 제시하지 못하며 단지 캐시를 사용하여 파이프라인 정지(stall)를 줄이고 성능을 높이는 기법만을 제공하였다.

주프로세서와 보조 프로세서를 외부 버스를 이용하여 연결시키는 방법은 프로세서 명령을 수행 중인 동안 주프로세서는 대기하여야 하고 보조 프로세서에서 필요한 데이터도 버스를 통하여 명령어와 같은 방식으로 전달하여야 하므로 수행 성능이 저하되며, 명령어 수준 병렬성(instruction-level Parallelism) 개념이 포함되어 있지 않다. 따라서 보조 프로세서에서 발생하는 예외 상황에 관한 특별한 대책도 포함되어 있지 않으며 단지 주프로세서와 연결된 버스를 통하여 상태를 통보할 뿐이다. 다중 프로세서 구조를 제시한 방법으로, 한 프로세서 내부에 하나 이상의 주프로세서들을 실장하고 다수의 프로세서들을 연결하는 방법이 제시되었다. 그러나 이 방법은 보조 프로세서를 효과적으로 공유할 수 있는 방법은 제시되지 않았다.

디커플링 개념을 적용한 방법으로 명령어 인출부와 주프로세서 및 보조 프로세서 수행부 사이에 명령어 큐를 위치시키는 방법이 있다. 그러나 이 방법은 다수의 주프로세서를 지원하지는 않으므로 공유하기 위한 기능은 없고 따라서 다수의 주프로세서로부터 요구받은 여러 명령어들을 동시 처리하기 위한 기능도 없다. 예외 상황에 대한 처리에 있어서도, 이 방법에서는 임시 레지스터들을 사용하였는데 임시 레지스터 수가 무한할 수 없으므로 비록 보조 프로세서 수행기가 디커플링되어 있어도 그 수행 시간의 범위가 상당히 제한되어 있어야 하며, 그렇지 않을 경우 예외 상황 발생시 복구할 수 없거나 프로세서 다른 부분들이 정지하여야 하는 문제점이 있었다.

본 발명이 이루고자 하는 기술적 과제

따라서, 본 발명은 주프로세서와 보조 프로세서 모두에 있어서 명령어 병렬성을 향상시키고 간단한 로직으로 예외 상황을 복구하며, 데이터의 종류에 따라 캐시의 효율을 저해하지 않고 데이터를 인출하고 저장할 수 있는 다수의 주프로세서 및 보조 프로세서를 갖는 프로세서의 구조 및 보조 프로세서 공유 방법을 제공하는데 그 목적이 있다.

상술한 목적을 달성하기 위한 본 발명에 따른 프로세서의 구조는 외부 메모리로부터 명령어와 데이터를 공급받아 프로그램을 수행하며 예외 상황에 대비하기 위한 대피 레지스터를 갖는 다수의 주프로세서와, 상기 주프로세서로부터 받은 명령어를 상기 주프로세서와 병렬 수행하며 대량의 데이터 처리시 알차 캐시 바이패스 기능이 있고 상기 다수의 주프로세서를 위한 레지스터 화일을 포함하는 다수의 보조 프로세서와, 상기 주프로세서 및 상기 보조 프로세서를 연결하는 내부 연결망을 포함하여 이루어진 것을 특징으로 한다.

또한 상술한 목적을 달성하기 위한 본 발명에 따른 보조 프로세서 공유 방법은 외부 메모리로부터 명령어 캐시/메모리 관리기 및 명령어 인출 및 검사 유닛을 통해 프로그램의 명령어들이 인출되는 단계와, 상기 인출된 명령어들이 보조 프로세서용인지 주프로세서용인지 검사하는 단계와, 상기 인출된 명령어의 검사 결과 주프로세서 명령어인 경우에는 주프로세서의 명령어 디코딩 유닛으로 명령어를 전달하여 내부 정보 형태로 변환하는 단계와, 상기 내부 정보 형태로 변환된 명령어를 주프로세서 명령어 미수 유닛으로 전달하는 단계와, 상기 명령어 미수 유닛으로 전달된 명령어에 대해 레지스터와 실행 유닛을 할당한 후 레지스터 화일을 포함한 실행 유닛들로 전달하는 단계와, 상기 실행 유닛으로 전달된 명령어에 대해 연산을 수행하는 단계와, 상기 연산을 수행한 후 필요한 데이터에 대해 데이터 인출/저장 과정을 수행하는 단계와, 상기 명령어를 외부로 저장하거나 외부로부터 데이터를 가져오는 경우 내부 데이터 캐시를 사용하거나 외부 메모리를 참조하여 데이터를 인출하는 단계와, 상기 인출된 명령어의 검사 결과 보조 프로세서 명령어인 경우 보조 프로세서의 명령어 큐로 전달하는 단계와, 상기 명령어 큐를 검사하여 명령어 미수 유닛에서 검사하여 동시에 수행 가능한 명령어들을 선택하는 명령어 스케줄링 단계와, 상기 명령어 미수 유닛에서 선택된 명령어에 대해 실행 유닛 및 레지스터 화일을 할당하는 단계와, 상기 명령어 미수 유닛에서 할당받은 정보에 따라 실행 유닛에서 연산을 수행하는 단계와, 상기 연산 수행시 예외 상황이 발생한 경우 주프로세서 상태를 되돌리기 위한 대피 레지스터 사용 과정 및 예외 상황 처리 과정을 수행하는 단계와, 상기 연산 수행 후 필요한 데이터에 대해 레지스터 화일을 사용하여 저장 또는 인출하는 단계를 포함하여 이루어진 것을 특징으로 한다.

본 발명의 구성 및 작용

이하, 첨부된 도면을 참조하여 본 발명을 상세히 설명하기로 한다.

도 1은 본 발명이 적용되는 보조 프로세서 공유 개념을 설명하기 위한 블록도로서, 하나 또는 그 이상의 주프로세서(101-1 내지 101-n)가 하나 또는 그 이상의 보조 프로세서(102-1 내지 102-m)를 공유하는 프로세서(11)의 구조를 나타낸다. 이때 보조 프로세서(102-1 내지 102-m) 또한 하나 이상이 될 수 있으며 보조 프로세서(102-1 내지 102-m)간에는 상호 작용이 없으므로 각각 병렬로 접속된다.

주프로세서(101-1 내지 101-n)는 외부 메모리(13)로부터 명령어와 데이터를 공급받아 프로그램을 수행하며 이때 보조 프로세서용 명령어를 인출한 경우 해당 보조 프로세서 명령어를 내부 연결망(12)을 통하여 적절한 보조 프로세서(102-1 내지 102-m)에게 보내준다. 명령어 수준 병렬성(instruction-level parallelism)을 최대한 살리기 위하여 주프로세서(101-1 내지 101-n)는 보조 프로세서(102-1 내지 102-m)가 명령어 수행을 완료하기를 기다리지 않고 명령어를 전달한 후 바로 다음 명령어들을 수행한다. 보조

프로세서(102-1 내지 102-m)는 주프로세서(101-1 내지 101-n)와 명령어를 병렬 수행하게 되며 수행이 완료되면 주프로세서(101-1 내지 101-n)에게 통보한다. 또한 보조 프로세서(102-1 내지 102-m)는 하나 이상의 주프로세서로부터 명령어를 전달받을 수 있으므로 수행에 필요한 레지스터 화일(register file)을 주프로세서의 수만큼 유지한다.

도 2는 본 발명에 따른 프로세서의 구조도이다. 프로그램의 흐름은 주프로세서(101-1 내지 101-n)가 관장하며 외부 메모리(도 1의 13)로부터 명령어 캐시/메모리 관리기(201) 및 명령어 인출 및 검사 유니트(202)를 통해 프로그램의 명령어들이 인출된다. 인출된 명령어들은 일단 보조 프로세서 용인지 검사하여 보조 프로세서 명령어인 경우 내부연결망(도 1의 12)을 통하여 보조 프로세서(102-1 내지 102-m)의 명령어 큐(208)로 보낸다. 주프로세서 명령어인 경우에는 주프로세서(101-1 내지 101-n)의 명령어 디코딩 유니트(203)로 명령어를 전달하여 명령어 종류와 필요한 자원을 비롯한 관련 정보들을 추출한다. 명령어 디코딩 유니트(203)에서 내부 정보 형태로 변환된 명령어는 주프로세서 명령어 미수 유니트(204)로 보내지며, 여기서 주프로세서(101-1 내지 101-n)의 내부 자원 상태를 고려하여 최대한 여러 명령어가 동시 수행될 수 있도록 레지스터와 실행 유니트를 할당하고 레지스터 화일(205)을 포함한 실행 유니트들(206-1 내지 206-m)들로 보내진다. 실행 유니트들(206-1 내지 206-m)은 주프로세서 명령어 미수 유니트(204)로부터 전달된 정보에 따라 사칙연산, 논리연산등의 동작을 수행하며 필요한 데이터는 레지스터 화일(205)을 사용하여 저장하고 인출한다. 명령어가 외부로부터 데이터 가져오거나 외부로 저장하는 명령일 경우 주프로세서 명령어 미수 유니트(204)는 저장/인출 유니트(207)로 정보를 보내며 저장/인출 유니트(207)는 외부로부터 데이터 레지스터 화일(205)로 데이터를 가져오거나 레지스터 화일(205)의 데이터를 외부로 저장한다. 이때 내부 데이터 캐시가 있는 경우 데이터 메모리 관리 기능을 거쳐 내부 데이터 캐시 사용되며 이 데이터 캐시에 해당 데이터 영역이 없을 경우(miss on the cache) 외부 메모리를 참조(access)하게 된다. 다수의 주프로세서(101-1 내지 101-n)들은 이와 같은 동작들을 서로 다른 주프로세서와 독립적으로 수행한다.

보조 프로세서(102-1 내지 102-m)는 명령어 큐(208)를 항상 감시(monitor)하고 있으며, 주프로세서(101-1 내지 101-n)로부터 관련 명령어가 입력되면 보조 프로세서 명령어 미수 유니트(209)가 동작한다. 보조 프로세서 명령어 큐(208)는 각 주프로세서당 하나씩 독립된 큐 구조를 지원하며, 따라서 주프로세서 수와 같은 n개의 큐 구조를 갖는다. 보조 프로세서 명령어 미수 유니트(209)는 각 주프로세서용 큐를 모두 검사하여 동시에 수행할 수 있는 명령어들을 선택한다. 각 주프로세서는 서로 독립된 프로그램 부분을 수행하고 레지스터 화일(205)을 따로 가지고 있으므로 보조 프로세서에서는 보조 프로세서 실행 유니트(211-1 내지 211-k)와 진행 중인 해당 보조 프로세서 레지스터 화일(210) 사용 가능성만 고려하면 된다. 즉, 주프로세서 또는 슈퍼스칼라 방식 고성능 프로세서들에 비해 단순한 제어회로만으로도 병렬성을 이룰 수 있다. 한 주프로세서만 보조 프로세서 명령어를 입력시킨 경우, 즉 주프로세서용 큐(208)가 하나만 유효한 명령어를 포함하고 있는 경우 보조 프로세서 명령어 미수 유니트(209)는 해당 큐의 여러 단을 동시에 검사하여 병렬 수행이 가능한 명령을 검색한다. 이 경우 하나의 주프로세서로부터 입력된 보조 프로세서 명령어간 병렬성으로 동시 수행될 명령어간 상호 의존성을 검사하여야 한다. 보조 프로세서 명령어 미수 유니트(209)에서 선택된 명령어들은 실행 유니트(211-1 내지 211-k)와 보조 프로세서 레지스터 화일(210)을 할당받는다. 실행 유니트(211-1 내지 211-k)는 보조 프로세서 명령어 미수 유니트(209)로부터 전달된 정보에 따라 사칙연산, 논리연산등의 동작을 수행하며 필요한 데이터는 해당 보조 프로세서 레지스터 화일(210)을 사용하여 저장하고 인출한다. 보조 프로세서 레지스터 화일 또한 주프로세서당 한 셋씩 있다. 보조 프로세서 명령어가 외부로부터 보조 프로세서로 데이터를 가져오거나 보조 프로세서 레지스터의 데이터를 외부로 저장하는 명령일 경우, 주프로세서 명령어 미수 유니트(204)가 주프로세서 저장/인출 유니트(207)를 제어하여 외부로부터 해당 주프로세서용 보조 프로세서 레지스터 화일(210)로 데이터를 가져오거나 이 레지스터 화일(210)의 데이터를 외부로 저장한다. 즉, 저장/인출 명령어는 보조 프로세서 데이터를 이동시키더라도 보조 프로세서에서 처리하지 않고 주프로세서에서 직접 처리하며 레지스터 화일만 보조 프로세서 레지스터 화일(210)을 직접 접근하여 사용한다. 이렇게 함으로써 보조 프로세서가 주프로세서의 메모리 관리 기능을 복제해 갖지 않으면서 외부 메모리에 대한 일관된 관리를 제공할 수 있다. 보조 프로세서(102-1 내지 102-m)가 여러 종류일 경우 각 보조 프로세서는 이상과 동일한 기능을 제공할 수 있지만 어떤 서로간 의존성을 고려할 필요 없이 여러 종류가 실장될 수 있다.

도 3은 다수의 명령어 큐를 대상으로 하는 명령어 스케줄링 개념을 설명하기 위한 블록도로서, 보조 프로세서의 명령어 미수 유니트(도 2의 209)내의 명령어 스케줄러(32)가 큐 구조들(31-1 내지 31-n)을 사용하여 다수의 보조 프로세서 명령어들을 동시 수행시키는 방법을 나타낸다. 이 큐 구조는 주프로세서당 하나씩 할당되므로 주프로세서와 같은 수(n)가 존재한다. 보조 프로세서의 명령어 스케줄러(32)는 n개의 큐 구조에서 첫 단들을 동시에 검사하며 이때 각 큐에 입력되어 있는 보조 프로세서 명령어들은 서로 독립적이다. 따라서 실행 유니트와 해당 레지스터 화일이 허용하는 한 여러 명령어들을 동시에 미수시킬 수 있다. 즉, 두 개 이상의 큐 구조(31-1 내지 31-n)에 동일한 보조 프로세서 실행 유니트(도 2의 211-1 내지 211-k)를 사용하는 명령어가 같은 위치의 단에 입력되어 있을 경우에는 이들 중 한 명령어만 미수되고 나머지는 실행 유니트(도 2의 211-1 내지 211-k)가 사용 가능해 질 때까지 대기해야 한다. 서로 다른 큐 구조에 있는 명령어는 서로 다른 레지스터 화일(도 2의 210)내의 셋을 이용하므로 레지스터에 대한 의존성은 없다. 그러나 동일한 큐 구조에서 앞서 미수된 명령어는 레지스터 화일(도 2의 210)내의 동일한 셋을 이용하므로 의존성이 있을 수 있다. 이 의존성은 한 사이클에 수행이 끝나지 않는 명령어인 경우에만 발생하며 특히 데이터 저장/인출 명령어가 수행되는 경우 발생할 수 있다. 데이터 저장/인출 명령어는 모두 주프로세서에서 수행되므로 주프로세서 명령어 미수 유니트(도 2의 204)는 보조 프로세서용 데이터 저장/인출 명령어를 수행할 때 보조 프로세서내 목표 레지스터 화일(도 2의 210)을 보조 프로세서에 통보하여야 하며 보조 프로세서 명령어 스케줄러(32)는 이 신호를 근거로 레지스터 사용 여부를 나타내는 자료 구조를 관리하여야 한다.

n 큐 구조 중 한 큐 구조(예를 들어 31-1)만 보조 프로세서 명령어를 가지고 있고 나머지(31-2 내지 31-n)는 비어 있는 경우에는 위와 같은 방법으로 여러 명령어를 동시 수행시킬 수 없다. 이러한 경우에는 보조 프로세서내 명령어 미수 유니트(도 2의 209)의 명령어 스케줄러(32)가 해당 큐 구조(31-1)를 수직적으로 검사하여 동일 큐 구조내에서 여러 명령어를 동시에 검사한다.

도 4는 보조 프로세서 공유를 위한 데이터 인출/저장 방법을 설명하기 위한 블록도로서, 보조 프로세서의 레지스터 화일(도 2의 210)을 각 주프로세서용 셋으로 분리하여(41-1 내지 41-n) 나타내었다. 저장/인출 유니트(43-1 내지 43-n)는 주프로세서내 저장/인출 유니트(도 2의 207)와 같은 것으로서 주프로세서 수만큼 존재한다. 캐시 메모리(44-1 내지 44-n)도 각 주프로세서가 독립적으로 가지고 있으며 최종적으로 전체 프로세서 외부로 나가는 인터페이스(45)는 모두 공유한다.

도시된 바와 같이 본 발명에 따른 보조 프로세서 공유 구조는 각 주프로세서(42-1 내지 42-n)마다 독립된 보조 프로세서 레지스터 화일(41-1 내지 41-n)과 캐시 메모리(44-1 내지 44-n)를 갖고 독립된 데이터 통로를 갖는다. 보조 프로세서를 위한 데이터 저장 또는 인출 명령어를 수행할 경우 해당 주프로세서(예를 들어 42-1)는 저장/인출 유니트(43-1)를 통하여 보조 프로세서의 레지스터 화일(41-1)을 액세스한다. 데이터를 외부로 저장하는 경우 저장/인출 유니트(43-1)는 해당 레지스터 화일(41-1)내 지정된 레지스터를 읽어 캐시 메모리(44-1)에 저장한다. 보조 프로세서용 데이터가 벡터 또는 어레이 형태로써 그 크기가 일반적인 데이터 보다 클 경우에는 캐시 메모리를 사용하지 않도록 데이터 형식(data type)을 지정할 수 있으며 이때 저장/인출 유니트(43-1)는 캐시 메모리를 바이패스(bypass)하도록 신호를 생성한다. 캐시 메모리를 바이패스하는 데이터는 바로 외부 인터페이스(45)를 통해 외부 메모리로 저장된다.

보조 프로세서 레지스터 화일(41-1)로 데이터를 인출해 올 경우 저장/인출 유니트(43-1)는 저장 때와 같이 데이터 형식에 따라 캐시 메모리(44-1)를 바이패스할 것인지 아닌지 결정하여 데이터 패스를 제어한다. 바이패스하는 경우 외부 인터페이스(45)를 통해 직접 레지스터 화일(41-1)로 데이터가 전송된다. 이와 같은 바이패스 기능은 작은 크기를 갖는 내부 캐시 메모리(44-1 내지 44-n)가 큰 데이터 이동에 감당하기 위해 많은 불력들을 교체해야 하는 부담을 덜어 주며 또한 교체된 불력들도 프로그램에서 곧 다시 사용될 확률이 높은 것들이므로 다시 캐시 메모리로 불려 들어오면서 보조 프로세서용 벡터 또는 어레이 형식 데이터를 내보내게 한다. 이와 같은 교체 반복 현상은 프로세서 성능에 매우 나쁜 영향을 미치므로 전송한 보조 프로세서 데이터 바이패스 기능이 효과가 크다.

또한 보조 프로세서 명령어 수행과 주프로세서 명령들이 동시에 병렬로 수행되기 때문에 성능상 이득은 크지만 보조 프로세서 명령 수행 도중 예외 상황(exception)이 발생하면 명령어 수행 이전 상태로 되돌아갈 수 있어야 한다. 이때 해당 보조 프로세서 명령어 이후의 주프로세서 명령어가 주프로세서에서 이미 수행되었거나 수행 중일 수 있으므로 주프로세서의 상태로 되돌릴 수 있어야 한다. 보조 프로세서 상태의 경우 예외 상황이 해당 명령어 수행 도중에 발생하는 것이므로 수행 완료로 인한 상태 변화가 있기 전이고, 따라서 단순히 해당 명령어 실행을 취소하기만 하면 된다. 이때 다른 주프로세서로부터 입력된 보조 프로세서 명령어들이 동시 수행되었을 수 있으나 이들은 도 2에 나타난 것처럼 구조상 서로 연관성이 없으므로 되돌리지 않아도 된다.

이와 같은 프로세서는 보조 프로세서와 주프로세서가 명령어 단위에서 동기되지 않고 각자 최대 성능을 낼 수 있는 방식으로 동작하고 프로그램 흐름은 모두 주프로세서에서 통제하므로 예외 상황 발생시 주프로세서 상태를 되돌리는(roll-back or recovery)것이 중요하다. 본 발명에서는 주프로세서의 레지스터 화일(도 2의 205)과 기타 필요한 레지스터들을 복사할 수 있는 대피 레지스터(shadow register) 화일을 각 주프로세서가 내장하여 상태 복귀시 사용한다.

도 5는 대피 레지스터의 사용 과정을 나타낸 흐름도이다. 대피 레지스터를 사용하기 위해 먼저, 전 보조 프로세서 명령어가 수행 중인지를 판단(51)한다. 판단 결과 전 보조 프로세서 명령어가 수행 중이면, 이미 주프로세서에 보조 프로세서 명령어 이전으로 돌아갈 수 있도록 대피된 레지스터 화일이 있는 것이므로 현재 사용 중인 레지스터 화일을 계속 사용하며 별도의 조치 없이 진행한다(52). 판단 결과 전 보조 프로세서 명령어의 수행이 종료된 경우에는 더 이상 보조 프로세서 예외 상황에 대비할 필요가 없으므로 사용 중인 레지스터의 최신 값으로 이전에 대피시킨 본 레지스터 화일(도 2의 205)로 복사하고(53) 사용 중인 레지스터 화일도 본 레지스터 화일(도 2의 205)로 전환(54)시켜 원래 상태로 돌아간다. 이후 새로운 보조 프로세서 명령어가 존재하는지를 판단(55)하여 새로운 보조 프로세서 명령어가 존재하지 않으면 현재 사용 중인 레지스터 화일을 계속 사용하며 별도의 조치 없이 진행한다(52). 만약 새로운 보조 프로세서 명령어가 존재한다면 해당 보조 프로세서 명령어 큐에 보조 프로세서 명령어를 입력(56)하고, 예외 상황에 대비하기 위하여 주프로세서 레지스터 화일(도 2의 205)을 대피 레지스터 화일로 복사하도록(57) 제어한다. 이 때 레지스터 화일과 함께 프로그램 수행에 필수적인 프로그램 카운터나 스택 포인터 또는 이와 유사한 레지스터들도 같이 대피된다. 이후 주프로세서는 복사된 대피 레지스터 화일(shadow register file)을 주 레지스터 화일로 전환(58)하여 사용하며 원래의 레지스터 화일은 해당 보조 프로세서 명령어가 보조 프로세서 명령어 큐로 입력되기 전 상태를 유지하므로써 대피 레지스터 사용 절차를 종료한다. 주프로세서의 인출 및 검사 유니트(도 2의 202), 명령어 디코딩 유니트(도 2의 203) 및 주프로세서 미수 유니트(도 2의 204)들은 협동하여 새 명령어가 미수되는 때 사이클마다 이 검사 동작을 반복한다.

이와 같이 하모리써 보조 프로세서 예외 상황이 발생한 경우 이에 대처하기 위해 보조 프로세서 명령어 미수시 프로그램 흐름에 따라 레지스터 화일과 필수 레지스터들을 전환하여 교차 사용할 수 있게 된다.

도 6은 보조 프로세서 예외 상황 처리 과정을 나타낸 흐름도이다. 보조 프로세서에서 명령어 수행중 예외 상황이 발생하면 보조 프로세서는 예외 상황을 발생시킨 명령어 수행을 취소(61)시키고 해당 명령어를 미수한 주프로세서에게 예외 상황을 통보한다(62). 보조 프로세서에서 명령어 수행중 예외 상황이 발생한 것을 주프로세서가 통보 받으면 일단 현재 진행 중인 수행을 중단해야 한다(63). 이때 해당 명령어를 미수하지 않은 다른 주프로세서들은 관계없이 진행 중인 동작을 계속 수행한다. 예외 상황을 통보 받은 주프로세서는 프로그램 흐름을 복구하기 위하여 프로그램 카운터를 보조 프로그램 미수 직전 상태로 되돌린다(64). 레지스터 화일도 보조 프로세서 명령어 미수시 대피시켰던 화일로 전환시킨다(65). 이후 주프로세서는 예외 처리 방식에 따라 예외 처리 루틴을 시작시킨다(66). 예외 처리 루틴이 종료하면 프로그램은 언제든지 예외 발생 상황 직전의 명령어부터 다시 수행할 수 있는 상태로 돌아간다.

이와 같이 본 발명에서는 보조 프로세서를 내장시키는 방식을 택하되 복수의 주프로세서가 보조 프로세서용 명령어 큐를 이용하여 효과적으로 공유하는 방법을 제시한다. 이러한 공유 방법은 보조 프로세서 예외 상황이나 보조 프로세서용 데이터 인출 및 저장, 그리고 보조 프로세서 내에서 명령어 병렬성을 살리도록 스케줄링하는 기능들에 대한 해결책이 있어야 실용성이 있다. 따라서 본 발명에서는 이들을 효과적으로

처리하기 위하여, 보조 프로세서 예외 상황에 대처하기 위하여 보조 프로세서 명령어 미수시 프로그램 흐름에 따라 레지스터 화일과 필수 레지스터들을 전환하여 교차 사용할 수 있는 방식을 제시한다.

또한 대량의 데이터를 집중 사용할 경우 작은 캐시 메모리를 바이패스할 수 있는 통로를 제공함으로써 전체 프로세서 성능을 저해하지 않으면서 보조 프로세서를 대량 데이터 접근 효율을 향상시킬 수 있도록 하였다. 여러 주프로세서가 공유하는 상황에서 보조 프로세서 내부 실행 유니트들을 병렬 수행시켜 성능을 향상시킬 수 있도록 복수의 큐간 명령어 스케줄링(inter-queue scheduling)과 큐 내부 스케줄링(intra-queue scheduling)을 동시에 제공하는 방식도 제시하고 있다. 이와 같이 하므로써 다수의 주프로세서를 갖는 프로세서에서 다양한 보조 프로세서들이 용이하게 공유될 수 있다.

발명의 효과

상술한 바와 같이 본 발명에 의하면 보조 프로세서를 공유하는 다수의 주프로세서들이 있는 환경에서 보조 프로세서와 주프로세서의 수행 효율을 극대화하여 보조 프로세서를 갖는 프로세서의 성능을 향상시키고 보조 프로세서 및 주프로세서를 독립적으로 변경, 개선 및 추가할 수 있도록 함으로써 프로세서 구조 개선과 구성 변경에 대한 유연성을 크게 증가시킬 수 있다. 또한 주프로세서가 모두 독립적으로 보조 프로세서를 가져야 하는 부담을 덜어 줌으로써 제한된 반도체 자원 내에서 다양한 보조 프로세서 기능을 제공할 수 있는 탁월한 효과가 있다.

(57) 청구의 범위

청구항 1

외부 메모리로부터 명령어와 데이터를 공급받아 프로그램을 수행하며 예외 상황에 대비하기 위한 대피 레지스터를 갖는 다수의 주프로세서와,

상기 주프로세서로부터 받은 명령어를 상기 주프로세서와 병렬 수행하며 대량의 데이터 처리시 일차 캐시 바이패스 기능이 있고 상기 다수의 주프로세서를 위한 레지스터 화일을 포함하는 다수의 보조 프로세서와,

서로 독립적으로 동작하는 상기 주프로세서 및 상기 보조 프로세서를 연결하는 내부 연결망을 포함하여 이루어진 것을 특징으로 하는 다수의 주프로세서 및 보조 프로세서를 갖는 프로세서의 구조.

청구항 2

제 1 항에 있어서, 상기 주프로세서는 외부 메모리로부터 입력된 명령어를 저장하는 명령어 캐시/메모리 관리기와,

상기 명령어 캐시/메모리 관리기로부터 명령어를 입력받아 프로그램 명령어를 인출하는 명령어 인출 및 검사 유니트와,

상기 프로그램 명령어를 내부 정보 형태로 변환하는 명령어 디코딩 유니트와,

상기 내부 정보 형태로 변환된 명령어들이 동시에 병렬 수행되도록 레지스터 및 실행 유니트를 할당하는 주프로세서 명령어 미수 유니트와,

상기 명령어 수행에 필요한 데이터를 저장하고 인출하는 레지스터 화일과,

상기 주프로세서 명령어 미수 유니트로부터 전달된 정보에 따른 동작을 수행하는 다수의 주프로세서 실행 유니트와,

상기 명령어를 외부 장치로 저장하거나 상기 명령어가 외부로부터 데이터를 가져오기 위한 정보를 관리하는 저장/인출 유니트를 포함하여 이루어진 것을 특징으로 하는 다수의 주프로세서 및 보조 프로세서를 갖는 프로세서의 구조.

청구항 3

제 1 항에 있어서, 상기 보조 프로세서는 상기 주프로세서의 명령어 인출 및 검사 유니트에서 인출된 명령어 중 보조 프로세서 명령어를 일시 저장하는 명령어 큐와,

상기 명령어 큐를 검사하여 동시에 수행 가능한 명령어를 선택하는 보조 프로세서 명령어 미수 유니트와,

상기 명령어 수행에 필요한 데이터를 저장하고 인출하는 주프로세서와 같은 수의 레지스터 화일과,

상기 명령어 미수 유니트로부터 전달된 정보에 따른 동작을 수행하는 다수의 보조 프로세서 실행 유니트를 포함하여 이루어진 것을 특징으로 하는 다수의 주프로세서 및 보조 프로세서를 갖는 프로세서의 구조.

청구항 4

제 3 항에 있어서, 상기 명령어 큐는 상기 주프로세서와 같은 수의 큐 구조를 각 주프로세서당 하나씩 독립적으로 지원하는 것을 특징으로 하는 다수의 주프로세서 및 보조 프로세서를 갖는 프로세서의 보조 프로세서 공유 방법.

청구항 5

외부 메모리로부터 명령어 캐시/메모리 관리기 및 명령어 인출 및 검사 유니트를 통해 프로그램의 명령어들이 인출되는 단계와,

상기 인출된 명령어들이 보조 프로세서용인지 주프로세서용인지 검사하는 단계와,

상기 인출된 명령어의 검사 결과 주프로세서 명령어인 경우에는 주프로세서의 명령어 디코딩 유니트로 명령어를 전달하여 내부 정보 형태로 변환하는 단계와,

상기 내부 정보 형태로 변환된 명령어를 주프로세서 명령어 미수 유니트로 전달하는 단계와,

상기 명령어 미수 유니트로 전달된 명령어에 대해 레지스터와 실행 유니트를 할당한 후 레지스터 화일을 포함한 실행 유니트들로 전달하는 단계와,

상기 실행 유니트로 전달된 명령어에 대해 연산을 수행하는 단계와,

상기 연산을 수행한 후 필요한 데이터에 대해 데이터를 저장/인출하는 단계와,

상기 명령어를 외부로 저장하거나 외부로부터 데이터를 가져오는 경우 내부 데이터 캐시를 사용하거나 외부 메모리를 참조하여 데이터를 인출하는 단계와,

상기 인출된 명령어의 검사 결과 보조 프로세서 명령어인 경우 보조 프로세서의 명령어 큐로 전달하는 단계와,

상기 명령어 큐에 대해 보조 프로세서의 명령어 미수 유니트에서 모든 명령어 큐를 수평적으로 검사하거나 하나의 명령어 큐를 수직적으로 검사하여 동시에 수행 가능한 명령어들을 선택함으로써 큐에 저장된 명령어 동시에 수행되도록 하는 명령어 스케줄링 단계와,

상기 명령어 미수 유니트에서 선택된 명령어에 대해 실행 유니트 및 레지스터 화일을 할당하는 단계와,

상기 명령어 미수 유니트에서 할당받은 정보에 따라 실행 유니트에서 연산을 수행하는 단계와,

상기 연산 수행시 예외 상황이 발생한 경우 주프로세서 상태를 되돌리기 위한 대피 레지스터 사용 과정 및 예외 상황 처리 과정을 수행하는 단계와,

상기 연산 수행 후 필요한 데이터에 대해 레지스터 화일을 사용하여 저장 또는 인출하는 단계를 포함하여 이루어진 것을 특징으로 하는 다수의 주프로세서 및 보조 프로세서를 갖는 프로세서의 보조 프로세서 공유 방법.

청구항 6

제 5 항에 있어서, 상기 대피 레지스터 사용 과정은

전 보조 프로세서 명령어가 수행 중인지를 판단하는 단계와,

상기 보조 프로세서 명령어의 수행 여부 판단 결과 전 보조 프로세서 명령어가 수행 중인 경우 현재 사용 중인 레지스터 화일을 사용하여 수행하는 단계와,

상기 보조 프로세서 명령어의 수행 여부 판단 결과 전 보조 프로세서 명령어의 수행이 종료된 경우에는 사용 중인 레지스터의 최신 값으로 이전에 대피시킨 본 레지스터 화일로 복사하는 단계와,

상기 사용 중인 레지스터 화일을 본 레지스터 화일로 전환시켜 원래 상태로 돌아가는 단계와,

새로운 보조 프로세서 명령어가 존재하는지를 판단하는 단계와,

상기 새로운 보조 프로세서 명령어의 존재 여부 판단 결과 새로운 보조 프로세서 명령어가 존재하지 않으면 현재 사용 중인 레지스터 화일을 사용하여 수행하는 단계와,

상기 새로운 보조 프로세서 명령어의 존재 여부 판단 결과 새로운 보조 프로세서 명령어가 존재하면 해당 보조 프로세서 명령어 큐에 보조 프로세서 명령어를 입력하는 단계와,

예외 상황에 대비하기 위하여 주프로세서 레지스터 화일을 대피 레지스터 화일로 복사하는 단계와,

프로그램 카운터 및 스택 포인터 등의 레지스터들을 대피 레지스터로 복사하는 단계와,

상기 복사된 대피 레지스터 화일을 주 레지스터 화일로 전환하여 사용하는 단계와,

원래의 레지스터 화일은 해당 보조 프로세서 명령어가 보조 프로세서 명령어 큐로 입력되기 전 상태를 유지하는 단계를 포함하여 이루어진 것을 특징으로 하는 다수의 주프로세서 및 보조 프로세서를 갖는 프로세서의 보조 프로세서 공유 방법.

청구항 7

제 5 항에 있어서, 상기 예외 상황 처리 과정은

예외 상황을 발생시킨 명령어 수행을 취소시키는 단계와,

해당 명령어를 미수한 주프로세서에게 예외 상황을 통보하는 단계와,

상기 예외 상황 통보를 받은 주프로세서가 현재 진행 중인 수행을 중단하는 단계와,

상기 예외 상황을 통보받은 주프로세서는 대피 레지스터 파일을 이용하여 프로그램 카운터 및 레지스터 화일을 보조 프로그램 미수 직전 상태로 복원하는 단계와,

예외 처리 방식에 따라 예외 처리 루틴을 처리하고 종료하는 단계로 이루어진 것을 특징으로 하는 다수의 주프로세서 및 보조 프로세서를 갖는 프로세서의 보조 프로세서 공유 방법.

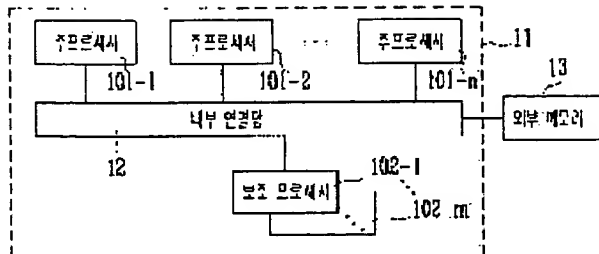
청구항 8

제 5 항에 있어서, 상기 데이터 인출/저장 방법은 보조 프로세서 명령어용 데이터의 종류에 따라 캐시 메모리

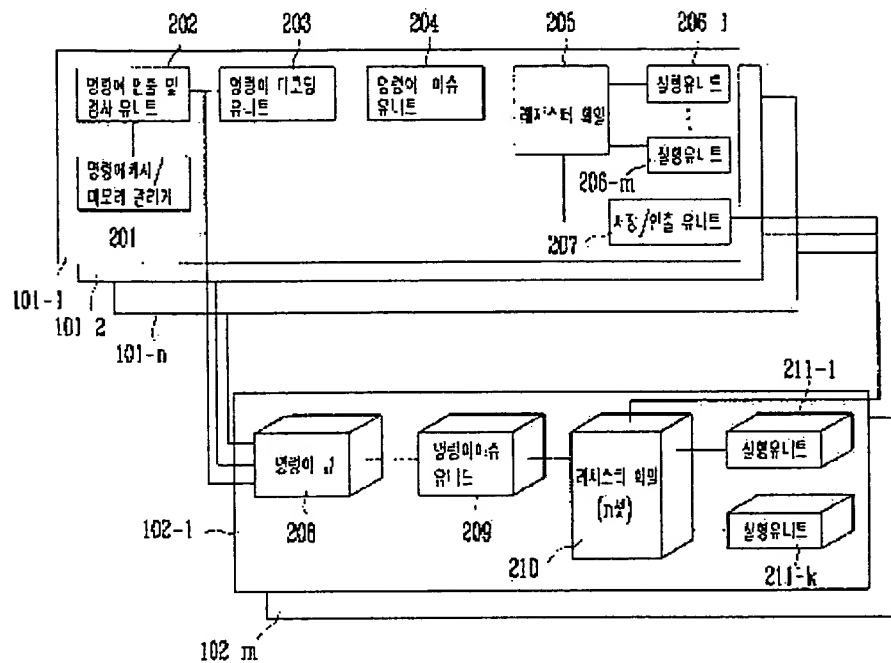
모리를 바이패스할 수 있도록 하는 것을 특징으로 하는 다수의 주프로세서 및 보조 프로세서를 갖는 프로세서의 보조 프로세서 공유 방법.

도면

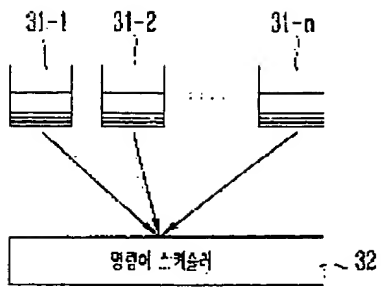
도면1



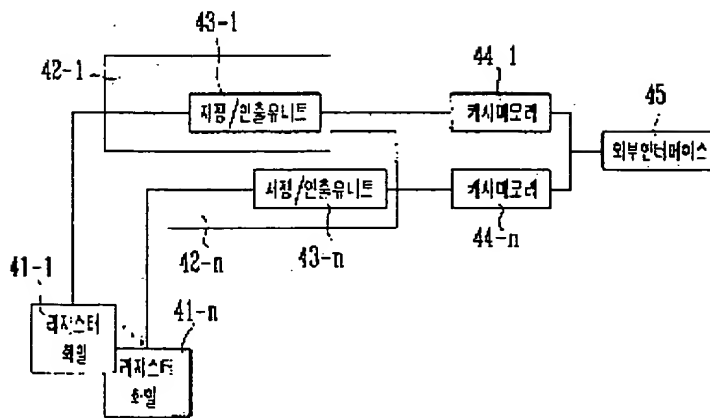
도면2



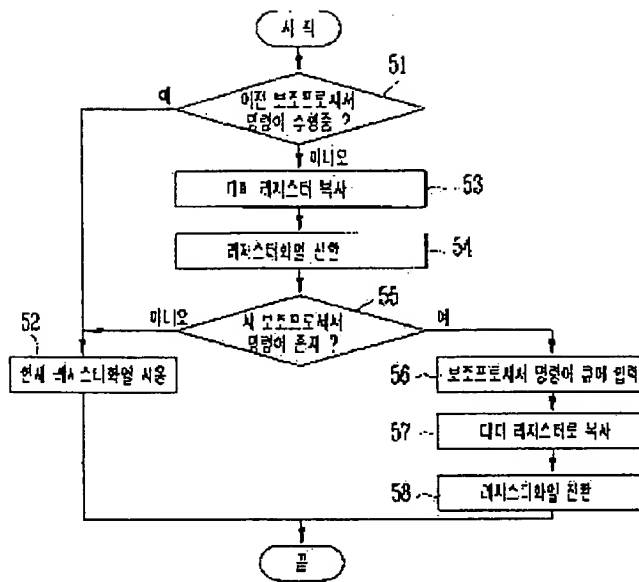
도면3



도면4



도면5



도면6

